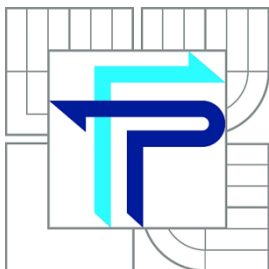


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
DEPARTMENT OF INFORMATION SYSTEMS

NÁVRH VYUŽITÍ XML KOMUNIKACE V INFORMAČNÍM SYSTÉMU FIRMY

XML COMMUNICATION USE DESIGN IN THE INFORMATION SYSTÉM OF A COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ BACHUREK

VEDOUCÍ PRÁCE

SUPERVISOR

ING JAN LUHAN

BRNO 2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Bachurek Ondřej

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh využití XML komunikace v informačním systému firmy

v anglickém jazyce:

XML Communication Use Design in the Information System of a Company

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Podle § 60 zákona č. 121/2000 Sb. (autorský zákon) v platném znění, je tato práce "Školním dílem". Využití této práce se řídí právním režimem autorského zákona. Citace povoluje Fakulta podnikatelská Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření "Licenční smlouvy" dle autorského zákona.

Seznam odborné literatury:

- BASL, J. Podnikové informační systémy. Podnik v informační společnosti. 1. vydání. Praha : Grada, 2002. 142 s. ISBN 80-247-0214-2.
- GILMORE, J. W. Velká kniha PHP a MySQL 5 : kompendium znalostí pro začátečníky i profesionály. 1. vydání. Brno : Zoner Press, 2007. 864 s. ISBN 80-86815-53-6.
- JANDO, J. ; POUR, J. ml. Technologie XML a její využití v e-obchodování (II) (XML technology and it's use in e-business). Systémová integrace. 2003, 9, 4, s. 24-41. ISSN 1210-9479.
- KOFLER, M. PHP 5 a MySQL 5 : průvodce webového programátora. 1. vydání. Brno : Computer Press, 2007. 607 s. ISBN 978-80-251-1813-9.
- MLÝNKOVÁ, I. et al. XML technologie : principy a aplikace v praxi. 1. vydání. Praha : Grada Publishing, 2008. 272 s. Dostupné z WWW:
<<http://books.google.com/books?id=oIsWTVQCsmIC&lpg=PA1&hl=cs&pg=PA1#v=onepage&q&f=false>>. ISBN 9788024727257.

Vedoucí bakalářské práce: Ing. Jan Luhan

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2010/2011.

L.S.

Ing. Jiří Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA
Dekan fakulty

V Brně, dne 31.05.2011

Abstrakt

Tato bakalářská práce se zabývá exportem dat z databáze MySQL elektronického obchodu TREF Josef Trojan do ekonomického a účetního programu Pohoda a propojení produktů s vyhledávacími portály za pomoci XML komunikace.

Abstract

This bachelor's thesis deals with the export of data from a MySQL electronic shop TREF Josef Trojan database into Economics and his well-being accounting software and networking products with search portals using XML communication.

Klíčová slova

Informační systém, informační technologie, XML, elektronický obchod, ekonomický a účetní systém Pohoda, vyhledávací portály.

Keywords

Information system, information technology, XML, electronic commerce, economic and financial welfare system Pohoda, search portals.

Bibliografická citace VŠKP dle ČSN ISO 690

BACHUREK, O. *Návrh využití XML komunikace v informačním systému firmy*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2011. 57 s. Vedoucí bakalářské práce Ing. Jan Luhan

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským).

V Brně, dne

.....

Podpis

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce panu Ing. Janu Luhanovi za odborné vedení, rady, věcné připomínky a ochotnou spolupráci při vzniku této bakalářské práce. Dále děkuji firmě TREF Josef Trojan za možnost zpracovávat tuto bakalářskou práci, poskytnuté informace a přístupy. Firma se mnou ochotně spolupracovala po celou dobu této práce.

Obsah

ÚVOD	9
1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE.....	10
2 TEORETICKÁ VÝCHODISKA PRÁCE.....	11
2.1 INFORMAČNÍ SYSTÉM FIRMY	11
2.1.1 Webové služby	12
2.1.2 Podnikový IS	13
2.2 XML.....	14
2.2.1 Základy jazyka XML.....	14
2.2.2 Základní struktura XML	17
2.2.3 Definice typu dokumentu	19
2.2.4 Jmenné prostory.....	21
2.2.5 XML schéma	22
2.2.6 Jednoduché datové typy.....	23
2.3 JAZYK PHP	24
3 ANALÝZA PROBLÉMŮ A SOUČASNÉ SITUACE	25
3.1 INTERNETOVÉ VYHLEDÁVAČE ZBOŽÍ	25
3.1.1 Vyhledávače zboží z pohledu zákazníka	25
3.1.2 Vyhledávače zboží z pohledu firmy	27
3.1.3 Základní struktura XML pro vyhledávač Zboží.cz	28
3.2 ANALÝZA IMPORTU/EXPORTU ZBOŽÍ Z EL. OBCHODU DO POHODY.....	31
4 VLASTNÍ NÁVRHY ŘEŠENÍ, PŘÍNOS NÁVRHŮ ŘEŠENÍ	34
4.1 NÁVRH ŘEŠENÍ VYHLEDÁVAČE ZBOŽÍ	34
4.2 NÁVRH ŘEŠENÍ IMPORTU/EXPORTU DAT IS	39
4.3 PŘÍNOS NÁVRHU ŘEŠENÍ.....	47
ZÁVĚR.....	48
SEZNAM POUŽITÉ LITERATURY	49
KNIŽNÍ ZDROJE.....	49
ELEKTRONICKÉ ZDROJE	49
SEZNAM OBRÁZKŮ.....	50
SEZNAM TABULEK.....	50
SEZNAM POUŽITÝCH ZKRATEK A POJMŮ	51

SEZNAM PŘÍLOH.....	1
PŘÍLOHA Č.1 - PHP SKRIPT PRO VYHLEDÁVAČ ZBOŽÍ ZBOŽÍ.CZ	1
PŘÍLOHA Č.2 - PHP SKRIPT PRO IMPORT/EXPORT DAT Z EL. OBCHODU DO POHODY.....	2

Úvod

V dnešním světě je používání informačních technologií (dále jenom IT) nezbytnou součástí podnikatelské sféry. Používání IT maximalizuje efektivitu, kvalitu a řízení firemních procesů v podniku. IT začínají poskytovat stále větší množství základních služeb, na kterých může koncový uživatel postavit řešení svého obchodního problému. Nejinak je tomu i v případě informačních systémů (dále jenom IS). Nedílnou součástí úspěšného podniku je integrace IS do procesu řízení. Velké množství dnes již úspěšných firem využívá IS jako svoji každodenní činnost.

Firma, se kterou spolupracuji, vlastní klientskou aplikaci elektronický obchod (dále jen el. obchod) a službu Pohoda na bázi IS, které by si přála propojit. Abychom tyto dva systémy propojili, bude zapotřebí řádně prostudovat základy jazyka XML a PHP, databázi MySQL patřící el. obchodu, a na základě těchto znalostí pak provést import/ export dat přes XML komunikaci. Další velmi potřebnou funkcí je propojení XML s internetovými nákupními průvodci, které nabízí užitečné informace o nabízeném zboží na internetu.

V této práci se budu věnovat základním charakteristikám IS, základním principům XML, a způsobu jeho propojení s nákupními průvodci a IS Pohoda. Návrh řešení využití XML v IS firmy zpracovávám pro firmu TREF Josef Trojan.

1 Vymezení problému a cíle práce

Firma TREF Josef Trojan využívá manažerského informačního systému POHODA pro účetní a ekonomickou evidenci své firmy.

V teoretické části této práce je potřeba se rámcově seznámit se základními pojmy podnikového IS, dále si vymezíme základy XML jazyka, jeho základní strukturu, definici typu dokumentu a XML schéma.

Na základě teoretických znalostí si v analýze probereme důležitost XML komunikace pro vyhledávače zboží z pohledu zákazníka a firmy. Dále provedeme analýzu aplikace el. obchodu na reálnou službu IS firmy TREF Josef Trojan.

Po analýze XML komunikace bude následovat její postupná implementace, která bude provedena pro internetový vyhledávač zboží serveru Zboží.cz. Dále bude proveden vlastní návrh exportu dat z MySQL databáze do databáze IS Pohoda.

Hlavním cílem této práce je úspěšná implementace XML pro vyhledávač zboží portálu Zboží.cz a taky úspěšné provedení importu/exportu dat mezi el. obchodem TREF Josef Trojan a IS Pohoda od firmy STORMWARE.

2 Teoretická východiska práce

2.1 Informační systém firmy

Jak již bylo zmíněno v úvodu, je potřeba předem vědět, co IS znamená, co obsahuje a jak vylepšit podnikové procesy informačního systému.

Informační systém (dále jenom IS) by měl být nedílnou součástí každé firmy. Pod názvem „informační systém“ si můžeme představit prakticky kterýkoliv systém, který pracuje s databází (podnikovými daty). Mezi informační systémy řadíme: od nejjednodušších, např. Microsoft Office Excel až po důmyslné podnikové informační systémy. Zde bychom se měli podívat na klasifikaci podnikových IS.

„Podle holisticko-procesní klasifikace tvoří podnikový IS:

***ERP**¹ jádro, zaměřené na řízení interních podnikových procesů,*

***CRM** systém obsluhující procesy směřované k zákazníkům,*

***SCM** systém řídící dodavatelský řetězec, jehož integrální součástí bývá **APS** systém sloužící k pokročilému plánování a rozvrhování výroby,*

***MIS** – manažerský informační systém, který sbírá data z ERP, CRM a APS/SCM systému (a samozřejmě také z externích zdrojů) a na jejich základě poskytuje informace pro rozhodovací proces podnikového managementu²”*

Podnikové informační systémy, které patří do kategorie ERP, se začaly zavádět počátkem devadesátých let a od té doby prošly velkým vývojem. Od technologie mainframe, přes klient-server až k současnému modulárnímu síťovému pojetí, které akceptuje SOA. Dodavatelé u těchto systémů navíc začínají respektovat mezinárodní standardy. (BASL, 2008)

¹ Seznam použitých zkratk, které jsou označeny tučně, jsou uvedeny na konci bakalářské práce

² SODOMKA, 2010, s. 77

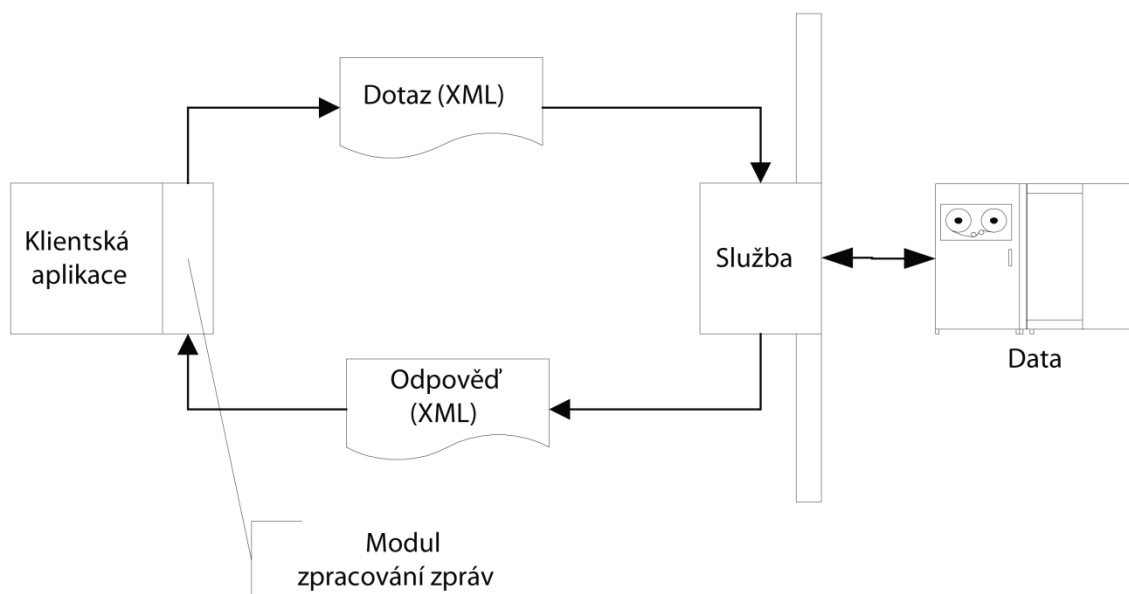
V dnešní době se objevuje stále více ERP systémů, které jsou postavené na bázi SOA a cloud computing.

„Cloud computing představuje model poskytování aplikací prostřednictvím Internetu v podobě služby. Ve srovnání s předchozími modely jej rozvíjí o možnosti grid computingu a clusterových řešení. Poskytování služeb touto formou je úzce spjato s uzavřením SLA. V každém případě nelze očekávat, že „v oblacích“ budou hromadně provozovány kritické podnikové aplikace.“³

Oba trendy jsou v dnešní době hojně akceptované technologie, které by mohly změnit spoustu věcí v praxi v podnikové informatice. Z technologie SOA se těší velké pozornosti **webové služby**. Ty jsou v dnešní době jedním z nejvíce vyskytovaných termínů na trhu v oblasti IS/IT.

2.1.1 Webové služby

Webová služba, na rozdíl od současného Webu, může být popsána jako sada programů, která interaguje po síti a u které během transakcí není potřeba účast člověka.



Obr. 1: Komunikace mezi klientskou aplikací a webovou službou
(BENDA, 2003, s. 403)

³ SODOMKA, 2010, s. 190

Na obrázku můžeme vidět jednoduchý příklad architektury webových služeb. V architektuře komunikují spolu dva účastníci, služba a klientská aplikace.

Jak služba, tak i klientská aplikace jsou aplikace, rozdíl je však v tom, že služba provádí na požádání nějakou akci (např. přebírají zaslaná data a vkládají je do databáze, poskytují data z registru apod.). Charakteristickou vlastností je naslouchání klientů. Naopak klientská aplikace žádá službu o nějakou akci (poskytnutí dat apod.). Jako službu bych proto zvolil IS Pohoda, kde provádí na požádání nějakou akci (např. přebírá z el. obchodu data a ty vkládá do databáze). A jako klientskou aplikaci by zvolil zmíněný el. obchod, který poskytuje data (v našem případě to budou data vygenerované PHP skriptem přímo z databáze).

Výhodou webových služeb je, že stávající aplikace (v našem případě el. obchod) je obalena webovou službou, aniž by nějak narušila chod el obchodu. Přenášená data za pomoci XML komunikace musí dodržovat přísnou sémantiku, tedy musí být zapsána pomocí jistých zásad.(BENDA, 2003)

2.1.2 Podnikový IS

V práci se nebudu zabývat vývojem celého IS, ale tzv. fází aplikační. Tzn., že se budu snažit o aplikaci, která spojí 2 IS dohromady přes tzv. XML komunikaci. V našem případě se bude jednat o podnikový informační systém. Mezi podnikové IS můžeme považovat (kromě již zmíněných v úvodu) taky ekonomické a účetní IS. Tyto IS se nezabývají interními procesy ve firmě, avšak „pouze“ ekonomickými a účetními operacemi. Mezi nejznámějšími IS v této oblasti patří Pohoda E1 od společnosti STORMWARE, dále pak Money od společnosti Cíglér software s.r.o..

Firma TREF Josef Trojan používá software Pohoda, se kterým si spravuje své účetnictví několik let. Je to velmi efektivní systém pro správu účetních a ekonomických operací, který je hojně využíván v malých, středních a větších firmách. Zabývá se množstvím agend, které si rozepíšeme v analýze importu/exportu zboží do IS Pohoda. (STORMWARE Pohoda, 2011)

2.2 XML

Od stručného návrhu řešení IS bych nyní přešel k základům jazyka XML, jeho syntaktického minima, definice typu dokumentu, XML schématu a základními datovými typy.

2.2.1 Základy jazyka XML

Jazyk **XML** je jednoduchým textově-založeným formátem zastupující strukturované informace: data, dokumenty, knihy, transakce, faktury a další.. XML jazyk je standardem, který je specifikovaný mezinárodním konsorciem W3C. Tato komunita se zabývá rozvojem webových standardů a jejím posláním je využít plného potenciálu pro vývoj protokolů a směrnic, které zajistí dlouhodobý růst webu. (QUIN).

O jazyku XML se mluví nejčastěji v souvislosti s internetovými stránkami, a považuje se za nástupce dnes hojně používaného jazyka **(X)HTML**. Výhodou XML oproti (X)HTML spočívá v tom, že si můžeme definovat své vlastní tzv. **tagy** (vysvětleno níže), které přesně určí význam daného textu v článku, což má velký význam ve vyhledávání informací. Dnešní svět internetu je přehlcen informacemi a nalézt konkrétní informací je stále těžší, mnohdy i nemožné. Tento problém nelze úplně vyřešit vyhledávacími servery, pokud jim nepomohou autoři stránek, kteří pomocí XML uloží do stránek mnohem více metainformací.(KOSEK, 1999)

XML patří mezi značkovací jazyky, které slouží pro značkování textu. Pod pojmem značkovací jazyk si můžeme představit libovolný text, který je podle jistých ustanovení označován. Tímto ustanovením je na mysli, že jsou použity „špičaté“ závorky, které jsou opět vyjádřené jako text. Tento text v závorkách značí význam textu, který je umístěný mezi závorkami, tzv. **tag**. Značky mají otevírací závorku (start-tag), např. **<nadpis>**, a zavírací závorku (end-tag), např. **</nadpis>**. Pokud nejsou tagy předem definované podle předpisu, můžeme mezi špičaté závorky umístit libovolně volitelný text, čili se jedná vlastně o metajazyk. Tato výhoda má velký význam především pro vyhledávání informací.(MLÝNKOVÁ, 2008)

Například pokud bychom chtěli vyjádřit nadpis článku: „Základy jazyka XML“, tak bychom tento nadpis zapsali `<nadpis>Základy jazyka XML</nadpis>`. Z kontextu je patrné, že text zapsaný mezi závorkami `<nadpis>` a `</nadpis>` znamená nadpis.

V základech jazyka XML jsme si definovali výhody a nevýhody XML oproti dnes hojně využívanému (X)HTML. Hlavním požadavkem je, aby byl celý dokument mezi počáteční a ukončovací tag. Všechny další tagy v dokumentu musí být taky párové, jak jsme si ukázali na jednoduchém příkladu části ceníku. Uveďme si nyní jednoduchý příklad části ceníku, který je uveden pomocí XML:

```
<vyrobek je="ano">  
  
<nazev>Rohlík</nazev>  
  
<cena>1,50</cena>  
  
</vyrobek>
```

(SOLDÁT, 2002)

Z příkladu můžeme vidět, že daný výrobek pravděpodobně existuje, jeho název je „Rohlík“ a stojí 1,50 (Kč). Dle tohoto zápisu pak není problém zadat serveru, aby nám našel informace o výrobku s názvem „Rohlík“ a jeho ceně. Veškeré výrobky potom lze jednoduše seřadit dle ceny. Na tomto příkladu můžeme sledovat obrovský posun oproti (X)HTML, kde bychom maximálně mohli hledat pouze ty stránky, které obsahují název „Rohlík“.

Na příkladu můžeme vidět, jak bychom mohli zapsat podobný příklad v jazyce (X)HTML:

```
<html>

<head><title>Ceník</title></head>

<body>

<table width="100%" border="1">

<tr><td bgcolor="red">Rohlík</td></tr>

<tr><td bgcolor="lightblue">1,50</td></tr>

</table>

</body>

</html>
```

Jazyk HTML připouští volnější syntaxi (neboli skladbu). Naopak XHTML má již striktnější pravidla pro zápis (např. rozlišuje malá velká písmena, nelze vynechat koncové závorky apod.). Z příkladu můžeme vidět, že v jazyce (X)HTML jsou tagy předem definované. Jejich význam nám z prvního pohledu nic moc neřekne, avšak pro zobrazení je tento jazyk velmi jednoduchým nástrojem. V XML jsou jména všech atributů a elementů citlivá na velikost písmen. Zvykem je psát vše malým písmenem. Taký je XML hodně citlivý na diakritiku, ovšem zde je to závislé na kódování celého dokumentu.

V jazyce (X)HTML jsou tagy předem definované. Tyto tagy však neznamenaají, co daný text znamená, avšak jak se má zobrazit. Na rozdíl od XML žádná podobná vazba neexistuje, proto musíme jednotlivé tagy zobrazovat jiným způsobem, a to pomocí stylů. Ve stylu již pak lze každý element definovat, jak se má zobrazit (barva, umístění, řez a velikost písma, zarovnání apod.). Nastýlované XML se nám potom může zobrazit přehledně pod sebou (jako např. v nějakém katalogu.).(KOSEK, 1999)

Stylování XML-dokumentu spočívá v tom, že styly jsou odděleny zvlášť v samotném souboru. Pokud tedy změníme určitý kód v souboru stylu, změní se tak vzhled ve všech XML-dokumentech (stejný princip platí i u (X)HTML, který používá k tomuto účelu kaskádové styly CSS). Pokud zobrazíte dokument bez požadovaného stylu, vyhodí Vám to data, které nejsou strukturované. Nové internetové prohlížeče však již strukturu XML dat dokážou sami nastýlovat. V praktické části si ukážeme XML soubor, který je nastýlovaný pomocí prohlížeče Google Chrome.

2.2.2 Základní struktura XML

Na začátku jsme psali, že jazyk XML je jednoduchým textově-založeným formátem zastupující strukturované informace. Tyto informace je však potřeba nějak zapsat. Na příkladu bych chtěl ukázat, jakým způsobem má vypadat základní struktura jazyka XML. Základním požadavkem na správně strukturovaný XML dokument je, aby byl celý uzavřen v jednom elementu. Celý dokument tedy musí být uzavřen mezi nějaký počáteční a ukončovací tag:

```
<cenik>  
  
...tělo dokumentu  
  
</cenik>
```

Pokud dále rozvinu tento základ, tak by mohl výsledný XML dokument vypadat např. následovně:

```
<?xml version="1.0" encoding="utf-8"?>

<cenik>

  <hlavicka>

    <firma>Pekárna, s.r.o.</firma>

    <adresa> Slaná 15, Rohlíkovice </adresa>

  </hlavicka>

  <vyrobky>

    <vyrobek je="ano">

      <nazev>Rohlík</nazev>

      <cena>1,50</cena>

    </vyrobek>

  </vyrobky>

</cenik>
```

V deklaraci dokumentu jsme si mohli všimnout kódování: za pomoci specifikace „version“

```
<?xml version="1.0" encoding="utf-8"?>

<cenik>

...tělo dokumentu

</cenik>
```

Dokument je tedy uložen v kódování „utf-8“, které má prvních 128 znaků stejných s ASCII tabulkou.(SOLDÁT, 2002)

2.2.3 Definice typu dokumentu

XML umožňuje popisovat pomocí jazyka **DTD** jednotlivé elementy dokumentu. Tato vlastnost má bezesporu obrovskou výhodu v tom, že obsahuje instrukce pro jiné aplikace. V praktické části definici typu dokumentu nepoužijeme, tuto kapitolu zde popisují z důvodu, že navazuje na jmenné prostory a následně i na XML Schema.

Nyní však přejdeme k definici DTD.

“V definici typu dokumentu zapsaného pomocí DTD se mohou vyskytovat:

- *deklarace typu elementu,*
- *deklarace seznamu atributů,*
- *deklarace entity*
- *deklarace notace⁴”*

Deklarace může být buď lokální (interní) anebo globální (externí), nebo obojí. Interní deklarace má vždy přednost před externí deklarací. Vždy se začíná konstrukcí DOCTYPE. Aby byl dokument **validní**⁵, musí odpovídat dle schématu.(MLÝNKOVÁ, 2008)

Pro vysvětlení deklarace DTD si uvedeme ukázkový příklad:

`<!ELEMENT jméno_elementu definice_elementu >`

Z ukázkového příkladu můžeme zkonstruovat interní deklaraci předchozího XML dokumentu. Prvky (definice elementu) se vkládají do kulatých závorek a oddělují se čárkou (znamená přesné pořadí elementů), anebo znakem „|“, který znamená „nebo“.

Pokud by měl být obsahem elementu pouze text, potom použijeme označení „(#PCDATA)“.

Uvedu zde příklad, jak by mohl takový XML dokument s interní DTD deklarací vypadat:

⁴ (MLÝNKOVÁ, 2008, s. 21)

⁵ **Validní** – platný podle jistých zásad, používá se např. u dokumentu

```

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE pekarstvi [

<!ELEMENT informace (firma, adresa, telefon?, email)>

<!ELEMENT firma (#PCDATA)>

<!ELEMENT adresa (#PCDATA)>

<!ELEMENT cenik (produkt*)>

<!ELEMENT produkt (nazev, cena)>

<!ELEMENT nazev (#PCDATA)>

<!ELEMENT cena (#PCDATA)>

]>

<pekarstvi>

  <informace>

    <firma>Pekárna, s.r.o.</firma>

    <adresa>Slaná 15, Rohlíkovice</adresa>

  </informace>

  <cenik>

    <produkt>

      <nazev>Rohlík</nazev>

      <cena>1,50</cena>

    </produkt>

    <produkt>

      <nazev>Houska</nazev>

      <cena>1,20</cena>

    </produkt>

  </cenik>

</pekarstvi>

```

(SOLDÁT, 2002)

Jak můžeme vidět, schéma deklarované pomocí DTD neumožňuje vytvářet datové typy, avšak definice vypadá poměrně jednoduše.

2.2.4 Jmenné prostory

Jednotlivé části dokumentu mohou vyhovovat různým DTD.

```
<cenik xmlns="zakladni_dtd"
      xmlns:pecivo="dtd_peciva">

  <pecivo:produkt>
    <pecivo:nazev>Houska</pecivo:nazev>
    <pecivo:cena>1,20</pecivo:cena>
  </pecivo:produkt>

  <pecivo:produkt>
    <pecivo:nazev>Rohlik</pecivo:nazev>
    <pecivo:cena>1,50</pecivo:cena>
  </pecivo:produkt>

  <pecivo:produkt pecivo:chutna="dobre">
    <pecivo:nazev>Kobliha</pecivo:nazev>
    <pecivo:cena>3,50</pecivo:cena>
  </pecivo:produkt>

</cenik>
```

Na prvním řádku můžeme vidět deklaraci základního DTD „<cenik xmlns="zakladni_dtd">“, kde „zakladni_dtd“ je adresa DTD ceníku, ve které jsou použity části vyhovující „dtd_peciva“. (SOLDÁT, 2002)

2.2.5 XML schéma

Jazyk DTD je pro účely XML schématu dostačující, ne však pro složitější aplikace, kde je potřeba si strukturu XML dokumentu vyjádřit přesněji. Oproti jazyku DTD obsahuje větší množství konstruktů a hlavně má silnou podporu datových typů. Většinu prvků jazyka DTD však zachovává.

V naší aplikaci se s XML schématem setkáme v praktické části u odkazovaných souborů, které jsou právě napsány pomocí XSD (**X**ML **S**chema **d**efinition - anglická zkratka, která označuje XML schéma).

Takto vypadá základní schéma:

```
<?xml version="1.0" encoding="utf-8"?>

<schema ...>

  <!--definice XML schématu>

</schema>
```

...které se dá přepsat následujícím způsobem:

```
<?xml version="1.0" encoding="utf-8"?>

<Korenovy_element

  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xsi:schemaLocation="http://www.priklad.cz/schemema1.xsd"

  xmlns="http://www.priklad.cz/MojeSchema">

  <!--XML dokument>

</korenovy_element>
```

Atribut xmlns:xsi určuje jmenný prostor instancí XML schémat (jehož identifikátorem je <http://www.w3.org/2001/XMLSchema-instance>). Podobně budeme odkazovat na jmenný prostor v naší práci (viz Obr. 13). (MLÝNKOVÁ, 2008)

2.2.6 Jednoduché datové typy

Jak jsem již psal, největší výhodou XML Schema je definice datových typů. Jednoduchý datový typ může být přiřazen atributu nebo elementu. Jak je tomu u většiny jazyků, u datových typů definujeme, zdali se jedná např. o řetězec znaků, celé číslo, reálné číslo apod. V následující tabulce napíšu nejzákladnější datové typy:

Tab. 1: Základní vestavěné datové typy (MLÝNKOVÁ, 2008, s. 53)

Název	Význam
string	Řetězec znaků
boolean	Logické hodnoty true a false, popř. 1 a 0
decimal	Kladné nebo záporné reálné číslo (např. -1.23,210,1267.5433)
float	32-bitové kladné nebo záporné reálné číslo vyjádřené pomocí mantisy a exponentu (např. -1E4, 1267.43233E12, 12, popř. speciální hodnoty 0, atd.)
double	64-bitové číslo se stejnými vlastnostmi jako float
dateTime	Datum a čas ve tvaru YYYY-MM-DDThh:mm:ss.ss, kde T je oddělovač, popř. se specifikací časové zóny
time	Čas ve tvaru gg:mm:ss.ss, popř. se specifikací časové zóny
date	Datum ve tvaru YYYY-MM-DD
gYearMonth	Měsíc v roce ve tvaru YYYY-MM
gYear	Rok ve tvaru YYYY
gMonthDay	Den v měsíci ve tvaru -MM-DD
gMonth	Měsíc ve tvaru --MM
gDay	Den ve tvaru --DD

V praktické části této práce budou datové typy dopředu definované, čili do struktury datových typů vůbec nebudeme zasahovat.

2.3 Jazyk PHP

Pro zvládnutí této práce je třeba ovládat základní úroveň PHP a databáze MySQL. Nebudu však vysvětlovat základní syntaxe jazyka PHP, protože rozsah této práce mi to neumožňuje. Uvedeme si zde alespoň základní informace o jazyku PHP.

PHP je skriptovací jazyk, který byl speciálně vytvořen pro tvorbu dynamických internetových stránek. Lze ho taky využít při tvorbě webových aplikací. Všechny operace jsou prováděny na straně serveru, uživatel vidí až konečný výsledek. PHP podporuje mnoho knihoven a taky databázových systémů, mezi které patří např. MySQL, PostgreSQL, Oracle, atd.. V kombinaci s operačním systémem Linux, zmíněných databázových systémů a webového serveru **Apache** je obvykle využíván k tvorbě webových aplikací. PHP existuje v několika verzích, pro práci s XML jazykem byla v mém případě použita PHP verze 5.1, v které byly odstraněny nedostatky z předchozích verzí. (KOFLEK, 2007)

PHP se vkládá přímo do (X)HTML stránky, kde se provede skript kdykoliv, když je stránka načtena. Tento skript se provede na straně serveru a na straně klienta se zobrazí vygenerovaný výsledek v (X)HTML kódu. Pomocí tohoto jazyka jsme schopni vygenerovat XML soubor s vhodnou strukturou, která musí splňovat webové standardy.

Mezi velké výhody PHP bych zařadil podporu pro velké množství již zmíněných databázových systémů. V případě naší firmy byla použita databáze MySQL el.obchodu. Podrobnější informace o jednotlivých použitých funkcích naleznete v praktické části.

3 Analýza problémů a současné situace

S jazykem XML se můžeme na internetu setkat prostřednictvím prezenčního formátu jazyka XHTML. Dále syntaxi XML využívá vektorový grafický formát SVG nebo jazyk pro definici uživatelského rozhraní v **RIA** prostředích, XAML v Silverlightu a nebo MXML ve Flashi. XML je taky využíván v tzv. RSS, což jsou formáty pro publikování přehledů nových článků.

Dále se můžeme s XML setkat prostřednictvím internetových vyhledávačů nebo v aplikační sféře např. těch programů, jejichž prostřednictvím lze provádět import/export dat. Právě těmito posledně zmíněnými operacemi se věnuji v této práci.

3.1 Internetové vyhledávače zboží

Jak již plyne z názvu, budu se zabývat vyhledáváním zboží na internetu. Všechny vyhledávací portály fungují na principu tzv. **XML feedu**, což znamená **URL** adresu datového XML souboru. V naší práci se budu zabývat službou Zboží.cz. Mezi další známé vyhledávače zboží v současnosti patří:

Bezvaceny.cz, cenyZbozi.cz, heureka.cz, hledejCeny.cz, hyperZbozi.cz, monitor.cz, naakup.cz, nejlepsiCeny.cz, nejNakup.cz, srovnave.cz, zbozi.Jyxo.cz

Všechny tyto portály jsem našel z průzkumu na internetu. Nejsou však řazeny podle kvality prodeje, avšak podle abecedy. Dle mého subjektivního názoru je nejlepším srovnávačem zboží na českém internetu portál Heureka.cz, kterou si trochu více popíšeme níže.

3.1.1 Vyhledávače zboží z pohledu zákazníka

Vyhledávací portály (nebo taky internetoví nákupní rádci) si v současnosti získaly velkou oblibu u zákazníků. Jedná se o portály, které jsou velmi jednoduché,

srozumitelné, přehledné pro zákazníka, a nabízejí velké množství informací o daném zboží, které zákazník hledá.

Z výzkumu, který proběhl v okruhu mých známých a přátel vyplývá, že zákazníci mohou nakupovat následujícím způsobem: nejprve se rozhodnou, které zboží chtějí koupit (např. hledají nějakou televizi nebo pračku). Dále postupují tak, že si porovnají nejrůznější parametry. Tyto parametry (informace) konzultují se svými známými, kteří už nákup přes vyhledávací portál uskutečnili. Pak si vyberou konkrétní výrobek, který má jasné typové označení. Na internetu vyhledají toto zboží za nejnižší cenu, která je na trhu dostupná. Přitom cena zboží je jeden z klíčových parametrů, které zákazníci zajímá. Další z klíčových parametrů pak zákazníka zajímají informace o daném produktu, ocenění od jiných zákazníků, jeho případná diskuze, a v neposlední řadě i ocenění samotného internetového obchodu. Vše je zaměřeno na rozhodnutí zákazníky, kteří již mají vybráno, našli zboží v nákupních rádčích (které je např. nejlevnější na trhu) a jsou tedy v podstatě rozhodnutí zboží koupit.

V tomto směru zákazníkům nabízí tyto služby již zmíněný vyhledávací portál Zboží.cz.

Zboží.cz je služba, která má na starosti vyhledání informací o nabízeném zboží. Jedná se o samotný přehled zboží jednotlivých obchodů, nikoliv však prodej zboží. Prodej zboží si zajišťují internetové obchody samy. Služba Zboží.cz je zdarma, má možnost i placeného režimu, tzv. služba Standart, avšak stále nenabízí tolik služeb, jako její konkurence.

Alternativou portálu Zboží.cz je Heureka.cz, která nabízí daleko více služeb, čili je zajímavou alternativou pro potenciální zákazníky. Heureka.cz nabízí oproti Zboží.cz katalog zboží, který obnáší vybírání produktů podle jednotlivých parametrů a specifikací produktů. Dále nabízí tzv. nákupního rádce, což je vlastně rada nebo diskuze ohledně produktů od samotných prodejců. Dále pak je zde diskuze, která se zabývá reálnými zkušenostmi zákazníků, zdali zboží má cenu koupit, nebo nemá, jeho hodnocení používání apod. Heureka.cz je zdarma, anebo si firmy mohou zaplatit placenou verzi, která obnáší cenu za tzv. „**proklik**“. Pokud si firma zaplatí cenu za

proklik, její produkty budou následně zařazeny do katalogu zboží Heureky.cz a firma bude mít tak možnost se zákazníky diskutovat ohledně daného produktu.

3.1.2 Vyhledávače zboží z pohledu firmy

Vyhledávací portály zboží nabízí obrovský přínos pro samotnou firmu. Po zaregistrování a přidání svého zboží do vyhledávacího portálu firma zjistí, zdali je cena jejího zboží konkurenceschopná. Takové srovnání cen zboží jistě firma ocení a bere tuto informaci za velmi přínosnou. Od ceny se pak dále rozvíjí firemní a marketingová strategie.

V případě naší firmy se po přidání zboží do nákupního rádce zjistilo, že firma má jedny z nejnižších cen zboží v daném odvětví. Důležité jistě je, aby taková firma dokázala využít příznivosti cen svého nabízeného zboží a snažila se nalákat co nejvíce zákazníků. Jedná se o tzv. marketingovou strategii, kde záleží na více aspektech, např. cenou zboží, jeho propagací, distribucí, apod. Zajištěním těchto aspektů se může firmě zvýšit celkový zisk a může se tak stát jedničkou na trhu.

3.1.3 Základní struktura XML pro vyhledávač Zboží.cz

V této části využijeme teoretických poznatků, kde byla zmíněna problematika XML a její základní struktura, ze které budeme vycházet. Výsledek pro Zboží.cz by měl tedy vypadat následovně:

```
<?xml version="1.0" encoding="windows-1250"?>

<SHOP>

<SHOPITEM>

<PRODUCT>Světélkující podložka pod myš</PRODUCT>

<DESCRIPTION>Fosforeskující okraj, nevyžaduje baterie.</DESCRIPTION>

<URL>http://obchod.cz/podlozky-pod-mys/fosfor</URL>

<ITEM_TYPE>new</ITEM_TYPE>

<DELIVERY_DATE>1</DELIVERY_DATE>

<IMGURL>http://obchod.cz/obrazky/podlozky-pod-mys/fosfor.jpg</IMGURL>

<PRICE>620</PRICE>

<PRICE_VAT>756</PRICE_VAT>

</SHOPITEM>

</SHOP>
```

Obr. 2: Základní struktura XML pro vyhledávač Zboží.cz (Nápověda Seznam.cz)

Tab. 2: Význam značek pro server Zboží.cz (Nápověda Seznam.cz)

<SHOP>	Kořenová značka (povinné)
<SHOPITEM>	Položka zboží (povinné)
<PRODUCT>	Hlavní název zboží (nepovinné, pokud jsou uvedeny elementy PRODUCTNAME a PRODUCTNAMEEXT)
<DESCRIPTION>	Popis zboží (povinné)
<URL>	Odkaz na stránku zboží (povinné)
<ITEM_TYPE>	Typ zboží, výchozí je „new“ (nepovinné)
<DELIVERY_DATE>	Doba vyřízení objednávky (nepovinné) - dodací doba musí být uváděna jako od přijetí platby do expedice zboží
<IMGURL>	Odkaz na obrázek zboží (nepovinné)
<PRICE>	Cena zboží bez DPH (nepovinné v případě, že je uvedený tag PRICE_VAT)
<PRICE_VAT>	Cena zboží s DPH (nepovinné, v případě, že je uvedený tag PRICE a tag VAT) – uvádí se 0.20 nebo 20, oba 2 zápisy jsou ekvivalentní a vyjadřují 20% sazbu

Tato tabulka byla použita ze specifikace služby Zboží.cz. Všechny uvedené tagy (značky) jsou párové, tzn., že obsahují start-tag a end-tag, který jsem již zmiňoval v teoretické části.

Mezi další důležité značky, které již nejsou povinné, patří:

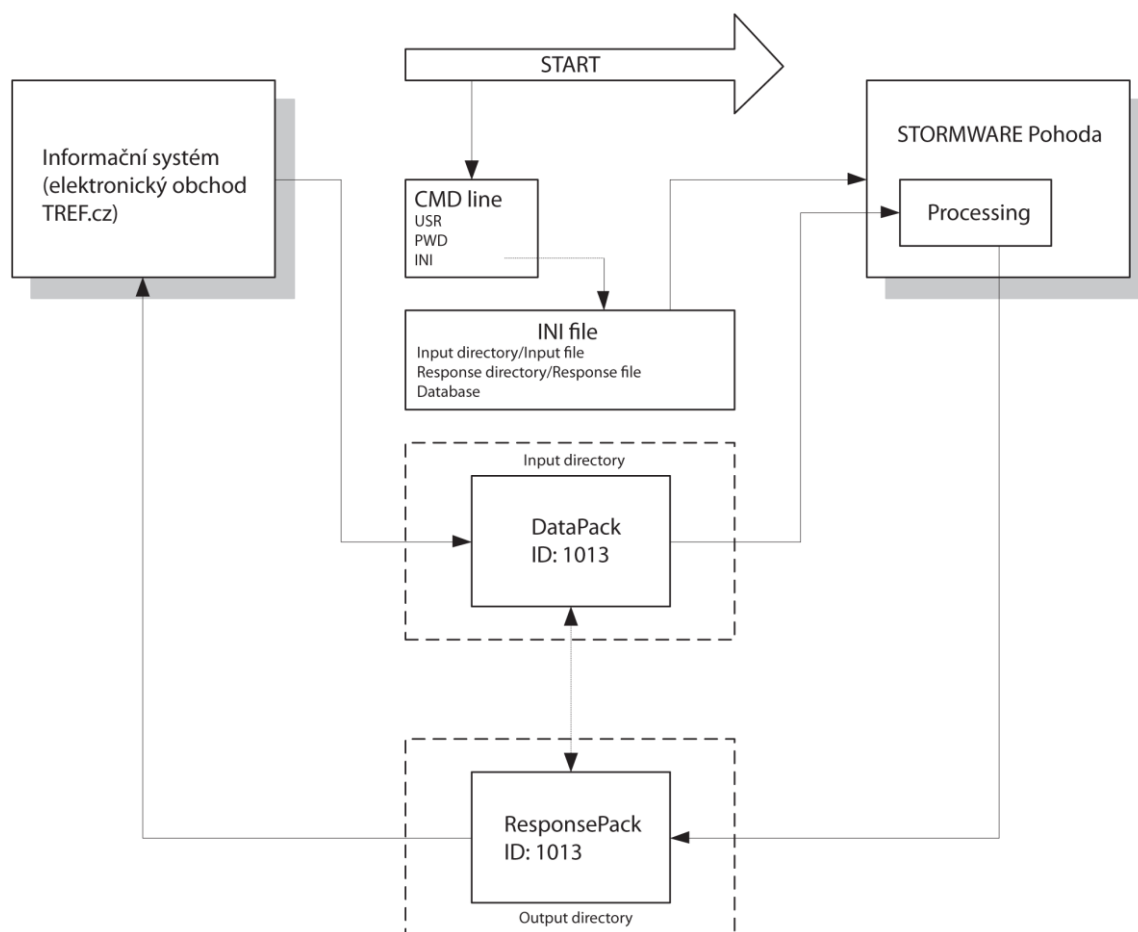
- <SHOP_DEPOTS> - Identifikátor kamenných poboček, ve kterých je položka skladem k okamžitému osobnímu odběru.
- <TOLLFREE> - Platí pouze pro placenou verzi Zboží.cz Standart. Znamená to, že proklik nebude zpoplatněn. Pokud tato informace nebude uvedena, tak po kliknutí na odkaz se automaticky u zpoplatněné verze strhává 1 Kč za proklik (Pozn. 1 Kč je současná hodnota za proklik u služby Standart).
- <MANUFACTURER> - název výrobce zboží
- <CATEGORYTEXT> - Zařazení zboží do kategorie, je třeba uvádět celou cestu ke zboží.
Jednotlivé kategorie by se měly oddělovat znakem „|“, povolený je znak „>“ za pomoci entity „>“.
- <EAN> - Jedná se o kód zboží EAN13, tedy čárový kód. Tento kód musí být validní včetně kontrolní číslice.

Dokument máme možnost zapsat dvěma způsoby: první vychází z generovaného XML souboru, který následně uložíme na server Zboží.cz a pak stačí pouze vypsát XML feed. Druhý způsob vychází z negenerovaného souboru, který sepíšeme ručně. Tento způsob je však hodně zdlouhavý a při množství tisíce produktů je téměř nemyslitelný. Proto budeme vycházet z prvního způsobu a XML soubor si vygenerujeme za pomoci PHP skriptu. K tomu je však zapotřebí znalost databáze a přístupových údajů. V praktické části této práce si ukážeme, jak takový PHP skript vypadá.

3.2 Analýza importu/exportu zboží z el. obchodu do Pohody

Od analýzy vyhledávačů zboží nyní přejdeme k té důležitější části, a tou je import/export dat zboží z elektronického obchodu do IS Pohoda. Základem tohoto systému je propracovaný adresář a řada agent, mezi něž patří např. majetek, pokladna, faktury, sklady apod., který bez problému zvládá obchodování na internetu, a to díky importu/exportu dat.

Pro ilustraci (viz Obr. 3) je obecný mechanismus automatizovaného XML importu/exportu. Jedná se o mechanismus, který je plně automatizovaný tak, aby mohlo docházet k importu/exportu dat.



Obr. 3: Obecný mechanismus automatizovaného XML importu/exportu (STORMWARE, 2011)

Pomocí tohoto mechanismu můžeme provádět operace, díky nimž může dojít k automatizované výměně dat mezi IS. Na obrázku (viz Obr. 3) je na jedné straně IS el. obchod TREF Josef Trojan a na druhé straně IS Pohoda. Jak bylo psáno již v úvodu, je třeba znát databázi el. obchodu, samozřejmě přístupové údaje, a v neposlední řadě musíme ovládat základní principy jazyků XML a PHP.

V práci jsem se rozhodl provést export dat z el. obchodu do programu Pohoda, ne však naopak. Bylo to kvůli ochraně dat firmy, která provádí v tomto programu veškeré účetní operace. Všechna tato data jsou velice citlivá a firma si je nepřála zveřejňovat. Dalším důvodem je fakt, že provádím pouze export dat z el. obchodu, tak si můžu zajistit jakoukoliv licenci programu Pohoda a zkoušet si procesy na jakékoliv verzi Pohody. Pro naše účely byla použita verze Pohody Start, která zvládá import/export dat, avšak v omezeném počtu operací, které však byly dostačující.

Hlavním důvodem, proč se takový import/export provádí je proces automatizace. Podobně, jak tomu bylo i u vyhledávačů zboží, je zapotřebí vygenerovat pomocí PHP skriptu XML soubor. Ten pomocí XML feedu nahrát do vašeho počítače, a poté provést import dat do programu Pohoda. Kdyby tento proces nebyl automatizovaný, všechna data bychom museli psát ručně. Proto každou operaci, kterou by firma provedla v el. obchodu, by musela ručně zapisovat do Pohody a to by zabralo spoustu času a úsilí. Snahou je proto co nejvíce tento proces zautomatizovat, aby firma ušetřila času a nákladů (čas jsou peníze).

Dalším důvodem těchto operací přes XML komunikaci je vyšší bezpečnost, neboť při komunikaci dochází k validaci a kontrole dat tak, jako kdyby se operace prováděly přímo uvnitř programu. Při operacích tak dochází ke kontrole vstupních údajů, proto vše musí být přesně napsané podle daných zásad, a pokud se takový soubor vygeneruje, tak musí projít tzv. **validátorem**.

Validátor zkontroluje schéma XML souboru, zdali máte vše podle zásad jazyka XML. V našem případě jsem použil STORMWARE Validator ve verzi 2.1, který mi umožnil bezpečně zkontrolovat vygenerovaný XML soubor. Bezpečnost této formy komunikace je srovnatelná s bezpečností ručního zápisu dat uvnitř programu Pohoda.

Mezi základní operace (nebo tzv. dokladové a seznamové agendy), které se dají provádět pomocí importu/exportu, patří:

Dokladové agendy:

Faktury, **objednávky**, nabídky, poptávky, interní doklady, pokladní doklady, příjemky, výdejky, převodky, výroba, prodejky.

Seznamové agendy:

Adresy, zásoby, zakázky, volitelné parametry, předkontace, uživatelské seznamy.

Tyto agendy jsou k dispozici ve verzi 1.0, anebo ve verzi 2.0, která je novější, a pro kterou jsem se rozhodl. V každé verzi jsou použité trochu jiné syntaxe. Abych mohl použít verzi 2.0, musel jsem si nainstalovat poslední lednovou verzi Pohody Start 2011. V seznamu agend jsem zvýraznil dokladovou agendu Objednávky, kterou jsem se rozhodl použít pro import dat do Pohody. Jak má přesně vypadat požadavek se strukturou přijaté objednávky s textovou a skladovou položkou, najdeme na webu firmy STORMWARE pod odkazem [Seznam dokladů pro import](#). Tuto strukturu je třeba při generování XML souboru přesně dodržet, aby se import dat zdařil.

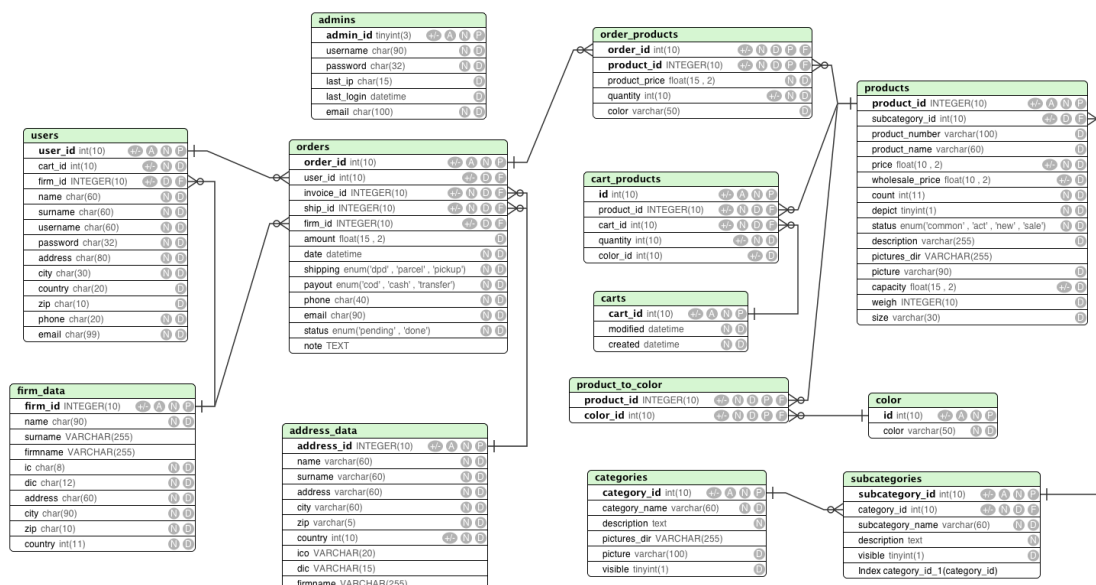
4 Vlastní návrhy řešení, přínos návrhů řešení

4.1 Návrh řešení vyhledávače zboží

V teoretické části této práce jsme si ukázali, jak takový XML soubor vypadá. V analytické části jsme si ukázali základní strukturu pro vyhledávač Zboží.cz, ze které budeme vycházet. Naším úkolem je pomocí PHP skriptu a MySQL databáze vygenerovat XML soubor, který následně nahrajeme na server, a poté použijeme jeho XML feed k vložení na server Zboží.cz.

Prvním krokem, který bychom měli učinit, je zajištění přístupu k databázi a **FTP přístupu** na server el. obchodu TREF Josef Trojan. Pokud máme k dispozici tyto přístupy, pak nám nic nebrání v tom, abychom si prošli databází el. obchodu.

Druhým krokem k úspěšné implementaci je řádné prostudování databáze příslušného el. obchodu. Veškeré podklady a přístupy el. obchodu mi ochotně poskytla firma TREF Josef Trojan. Na základě těchto podkladů jsem si vytáhl schéma databáze a zakreslil jej v příslušném programu. Na obrázku (viz Obr. 4) můžeme vidět příslušné schéma databáze:



Obr. 4: Schéma databáze firmy TREF Josef Trojan (vlastní zpracování)

Na základě schématu (viz Obr. 4) přesně dokážeme určit, kde se daná tabulka nachází, a podle ní pak přesně naprogramovat výsledný PHP skript. Nyní již máme veškeré přístupy a schéma databáze, tak se můžeme pustit do programování. Na obrázku (viz Obr. 5) můžeme vidět část PHP skriptu, pomocí kterého vygenerujeme XML soubor:

```
1 <?php
2
3 // natažení knihoven eshopu
4 require_once("../functions/global_fns.php");
5 require_once("../functions/eshop_fns.php");
6
7 // připojení k databázi
8 $conn = db_connect();
9
10 // výběr všech produktů z databáze
11 $query = "SELECT * FROM products";
12
13 // provedení dotazu
14 $result = $conn->query($query);
15
16 // vytvoření otevření souboru, který je připraven pro zápis XML
17 $fp = fopen('xml_feed_zbozi.xml', 'w');
18
19 fwrite($fp, "<?xml version=\"1.0\" encoding=\"windows-1250\"?>\r\n");
20 fwrite($fp, "<SHOP>\r\n\r\n");
21
22 // vybrání řádku z tabulky
23 while($row = $result->fetch_assoc()) {
24     extract($row);
25     if($description == NULL || !$depict)
26         continue;
```

Obr. 5: 1. část PHP skriptu pro vyhledávač Zboží.cz (vlastní zpracování)

Rozsah této práce mi neumožňuje dopodrobna okomentovat celý kód, proto se budu vyjadřovat pouze k nejdůležitějším syntaxím. Na začátku je třeba natáhnout všechny knihovny, bez kterých by se skript nespustil. Tyto knihovny již na serveru byly a slouží k přihlášení do databáze včetně přístupových kódů, které nemohou být zveřejněné. V řádce 8 (viz Obr. 5) se volá funkce pro vytvoření spojení k databázi, která se však nachází v jiném souboru.

Dále stojí za zmínku výběr tabulky „products“, která obnáší dané informace o produktech v el. obchodu. Zajímavou funkcí je funkce „fopen“, která má příznak „w“, což slouží k otevření souboru **pouze** pro zápis. Pokud daný soubor, který se v této funkci nachází, neexistuje, pokusí se ho daná funkce vytvořit. V závorkách funkce se může nacházet jak název samotného souboru, tak i příslušná cesta k souboru. Další

funkcí je funkce „fwrite“, pomocí které zapisujeme obsah daného řetězce do souboru. Jak si můžeme všimnout na 19. a 20. řádku (viz Obr. 5), tak daný řetězec obsahuje syntaxe, které byly popsány ve specifikaci XML na serveru Zboží.cz. Další věcí, která je nutná, je vybrání řádku z tabulky databáze a to za pomoci funkce „extract“, která extrahuje pole, z klíčů udělá proměnné, a do proměnných se dosadí hodnoty. Proto potom můžeme provést zápis následujícím způsobem:

```
fwrite($fp, "<PRODUCT>".$product_name."</PRODUCT>\r\n");
```

v opačném případě bychom museli provést zápis následujícím způsobem:

```
fwrite($fp, "<PRODUCT>$row['product_name']</PRODUCT>\r\n");
```

V případě, že popis produktu bude prázdný nebo produkt nebude obsahovat žádný obrázek, se nebude pokračovat v provádění PHP skriptu, a skript se poté ukončí. Co se týče proměnné subkategorie a kategorie, tak se zde získávají informace o subkategorii a kategorii produktu.

```
32
33     $count <= 0 ? $availability = "-1" : $availability = "0";
34
35     fwrite($fp, "<SHOPITEM>\r\n");
36     fwrite($fp, "<PRODUCT>".$product_name."</PRODUCT>\r\n");
37     fwrite($fp, "<DESCRIPTION>".$product_name."</DESCRIPTION>\r\n");
38     fwrite($fp, "<URL>http://www.tref.cz/product_detail.php?prodid=".$product_id."</URL>\r\n");
39     fwrite($fp, "<ITEM_TYPE>new</ITEM_TYPE>\r\n");
40     fwrite($fp, "<DELIVERY_DATE>".$availability."</DELIVERY_DATE>\r\n");
41     fwrite($fp, "<IMGURL>http://tref.cz/pictures/".$pictures_dir."/thumbs/thumb_".str_replace(" ", "%20", $picture)."</IMGURL>\r\n");
42     fwrite($fp, "<PRICE_VAT>".$price."</PRICE_VAT>\r\n");
43     fwrite($fp, "<CATEGORYTEXT>".$category["category_name"]." | ".$subcategory["subcategory_name"]."</CATEGORYTEXT>\r\n");
44     fwrite($fp, "</SHOPITEM>\r\n\r\n");
45 }
46
47 fwrite($fp, "</SHOP>\r\n");
48 ?>
```

Obr. 6: 2. část PHP skriptu pro vyhledávač Zboží.cz (vlastní zpracování)

V dalších řádcích zdrojového kódu (viz Obr. 6) je opět použita funkce „fwrite“ a vše je zapsáno pomocí specifikace Zboží.cz, která se musí přesně dodržet. Jak si můžeme všimnout, tak nejsou použity všechny tagy, které jsme si popsali v analýze problémů. Použity jsou pouze ty tagy, které se daly použít.

Nyní již máme PHP skript hotový, stačí jej tedy vložit na server a spustit tento skript přes prohlížeč. Pokud skript nevyhodí žádnou chybu, tak vše proběhlo v pořádku a vytvoří se soubor „xml_feed_zbozi.xml“. Část tohoto souboru vypadá následovně:

```

▼<SHOP>
  ▼<SHOPITEM>
    <PRODUCT>Sud 120 l zelený</PRODUCT>
    <DESCRIPTION>Sud 120 l zelený</DESCRIPTION>
    <URL>http://www.tref.cz/product_detail.php?prodid=2</URL>
    <ITEM_TYPE>new</ITEM_TYPE>
    <DELIVERY_DATE>0</DELIVERY_DATE>
    ▼<IMGURL>
      http://tref.cz/pictures/2_sud_120_l_zeleny/thumbs/thumb_upr.sud%20120%20l%20zeleny.JPG
    </IMGURL>
    <PRICE_VAT>408.00</PRICE_VAT>
    <CATEGORYTEXT>plastové sudy a kádě | sudy zaklapovací</CATEGORYTEXT>
  </SHOPITEM>
  ▼<SHOPITEM>
    <PRODUCT>Sud 30 l šroubovací</PRODUCT>
    <DESCRIPTION>Sud 30 l šroubovací</DESCRIPTION>
    <URL>http://www.tref.cz/product_detail.php?prodid=643</URL>
    <ITEM_TYPE>new</ITEM_TYPE>
    <DELIVERY_DATE>0</DELIVERY_DATE>
    ▼<IMGURL>
      http://tref.cz/pictures/643_sud_30_l_sroubovaci/thumbs/thumb_upr.sud%2030%20l%20sroub.JPG
    </IMGURL>
    <PRICE_VAT>237.00</PRICE_VAT>
    <CATEGORYTEXT>plastové sudy a kádě | sudy šroubovací</CATEGORYTEXT>
  </SHOPITEM>
  ▼<SHOPITEM>
    <PRODUCT>Borovice T13 světlá</PRODUCT>
    <DESCRIPTION>Borovice T13 světlá</DESCRIPTION>
    <URL>http://www.tref.cz/product_detail.php?prodid=32</URL>
    <ITEM_TYPE>new</ITEM_TYPE>
    <DELIVERY_DATE>0</DELIVERY_DATE>
    ▼<IMGURL>
      http://tref.cz/pictures/32_borovice_t13_svetla/thumbs/thumb_borovice%20svetla%20T13%20I.jpg
    </IMGURL>
    <PRICE_VAT>22.00</PRICE_VAT>
    <CATEGORYTEXT>polyst. stropní kazety | kazety extrudované</CATEGORYTEXT>
  </SHOPITEM>

```

Obr. 7: Část výsledného XML souboru (vlastní zpracování)

Jak můžeme vidět (viz Obr. 7), XML kód je napsaný přesně podle specifikace Zboží.cz. Tento soubor ve skutečnosti obsahuje přes tisíc produktů a přes osm tisíc řádků XML kódu, proto psát ručně všechny tyto produkty by nám zabralo i několik týdnů.

Nyní je vše přichystáno k vložení na server Zboží.cz. Stačí se zde zaregistrovat a vyplnit údaje firmy včetně **URL** adresy vygenerovaného souboru (XML feedu). Pokud počkáme několik dní na zpracování tohoto souboru (několik dní z toho důvodu, že se využívá neplacená verze), tak se dočkáme přidání produktů na server. Výsledek pak při zadání dotazu „Sud 120 l zelený“ bude vypadat následovně:

Internet Firmy Mapy Slovník Zboží Obrázky Encyklopedie Seznam











zboží.cz

Nové
[Bazarové](#)

[Dle nejlepší shody](#)
[Od nejnižšího](#)
[Od nejdražšího](#)

- vyberte kraj -


[Upravit cenu](#)
[Dostupnost: Vše](#)
[Způsob platby: Vše](#)

	Vlko na sud 120 l zelený 78,- Kč s DPH Skladem v e-shopu	TREF informace o obchodu
	[800024] Sud PE zelený s víkem 120 l (Produkty pro zahradu) 402,- Kč s DPH Skladem v e-shopu	EICO PLUS informace o obchodu
	Sud 120 l zelený 408,- Kč s DPH Skladem v e-shopu	TREF informace o obchodu
	Sud 120 l šroubovací zelený 440,- Kč s DPH Skladem v e-shopu	TREF informace o obchodu
	Sud zelený s víkem - 120 l 470,- Kč s DPH Do týdne v e-shopu	Zahradnidum.cz informace o obchodu
	Sud zelený s víkem - 120 l 518,- Kč s DPH Skladem v e-shopu	Zahradní centrum Jin... informace o obchodu
	Sud zelený s víkem - 120 l 518,- Kč s DPH Neznámá dostupnost	Farmářka.cz informace o obchodu
	Sud zelený s víkem - 120 l (500241 sud soudek na vodu) 530,- Kč s DPH Více než týden, na objednávku: Upice	MELICHAR CZ s.r.o. informace o obchodu
	Sud zelený s víkem - 120 l 585,- Kč s DPH Skladem v e-shopu	zahradou.cz informace o obchodu
	Sud zelený s víkem - 120 l 585,- Kč s DPH Neznámá dostupnost	Zahradní centrum v D... informace o obchodu

Zobrazujeme 1 - 10 z 12 nalezených

1 2 [Další](#)

[Přidat obchod](#) - [Administrace obchodu](#) - [Přidání inzerce](#) - ☒ [Statistika dotazu "Sud 120 l zelený"](#) - [mPeníze pro internetové obchody](#)
© 1996-2011 Seznam.cz, a.s.
[Mobilní verze](#) - [Nápověda](#) - [Kontakt](#)

výsledky dodává
SEZNAM.CZ


Obr. 8: Výsledek vyhledávání zboží na serveru Zboží.cz (vlastní zpracování)

Na obrázku (viz Obr. 8) můžeme vidět cenové srovnání zboží, které je seřazené od nejnižší ceny. Jak jsem již uvedl v analýze problémů, tak firma TREF Josef Trojan má opravdu jedny z nejnižších cen na serveru Zboží.cz.

4.2 Návrh řešení importu/exportu dat IS

Při návrhu řešení importu/exportu dat IS se k úspěšné implementaci dají uplatnit první dva kroky, které byly popsány v návrhu řešení vyhledávače Zboží.cz. Stejně tak můžeme vycházet ze stejných přístupů k databázi a FTP serveru, a proto se rovnou můžeme pustit na PHP skript:

```
1 <?php
2
3 // natažení knihoven eshopu
4 require_once("../functions/global_fns.php");
5 require_once("../functions/eshop_fns.php");
6
7 // připojení k databázi
8 $conn = db_connect();
9
10 // výběr všech objednávek z databáze
11 $query = "SELECT *,
12           DATE_FORMAT(date, '%Y%m%d') AS numberOrder_date,
13           DATE_FORMAT(date, '%Y-%m-%d') AS date_notime
14           FROM orders";
15
16 $result = $conn->query($query);
17
18 // vytvoření otevření souboru, který je připraven pro zápis XML
19 $fp = fopen('xml_orders2.xml', 'w');
20
21 // zápis hlaviček xml dokumentu
22 fwrite($fp, '<?xml version="1.0" encoding="Windows-1250"?>
23           <dat:dataPack id="ob001" ico="12345678" application="StwTest"
24           version = "2.0" note="Import Objednávky"
25           xmlns:dat="http://www.stormware.cz/schema/version_2/data.xsd"
26           xmlns:ord="http://www.stormware.cz/schema/version_2/order.xsd"
27           xmlns:typ="http://www.stormware.cz/schema/version_2/type.xsd">');
```

Obr. 9: 1. část PHP skriptu pro import/export dat do Pohody (vlastní zpracování)

Jak můžeme vidět na obrázku (viz Obr. 9), začátek PHP skriptu je podobný jako v předchozím návrhu řešení. Jsou zde použity stejné knihovny a nejinak je tomu i s připojením k databázi. Abychom provedli export objednávek, vybereme z MySQL databáze tabulku „orders“. Jelikož byl špatně nastavený formát data, bylo potřeba ho upravit. I zde jsem vytvořil soubor, který je připraven pro zápis XML pomocí příznaku „w“, čili „write“.

Soubor je rozlišen až v zápisu hlavičky XML dokumentu. Strukturu XML souboru objednávky nalezneme na stránce firmy STORMWARE a je třeba jí striktně dodržovat. Od řádku 22 na obrázku (viz Obr. 9) můžeme vidět hlavičku výsledného

XML souboru, která je ve verzi 2.0 s id „ob001“ a kde bylo použito všeobecné označení firmy pomocí IČO „12345678“ (nejedná se však o IČO firmy TREF Josef Trojan).

```
28 // inicializace pomocných proměnných
29 $pocitadlo = 1;
30
31 // zpracování objednávek
32 while($row = $result->fetch_assoc()) {
33
34 // převod počítadla objednávek na 3-místné číslo
35 if($pocitadlo < 10)
36     $pocitadlo_text = "00".$pocitadlo;
37 elseif($pocitadlo < 100)
38     $pocitadlo_text = "0".$pocitadlo;
39 else
40     $pocitadlo_text = $pocitadlo;
41
42 // převod způsobu dopravy na české ekvivalenty
43 switch($row['payout']) {
44     case "cod":
45         $payment = "na dobírku";
46         break;
47     case "cash":
48         $payment = "hotově";
49         break;
50     case "transfer":
51         $payment = "převodem";
52 }
53
54 // převod proznaku stavu objednávky
55 switch($row['status']) {
56     case "done":
57         $delivered = "1";
58         break;
59     case "pending":
60         $delivered = "0";
61 }
```

Obr. 10: 2. část PHP skriptu pro import/export dat do Pohody (vlastní zpracování)

Na dalším obrázku (viz Obr. 10) jsem použil pomocí podmínky IF jednoduché počítadlo objednávek, aby se v objednávkovém čísle zobrazil mj. kód jako tří místné číslo, začínající číslem 001. Dále bylo třeba ošetřit status, kde máme dva stavy. Buď je objednávka doručena anebo není. V šabloně u tagu „delivered“ je datový typ „boolean“, čili dva stavy 0 a 1, stavu s názvem „done“ jsem přiřadil hodnotu 1 a stavu „pending“ jsem přiřadil hodnotu 0.

```

71 // zápis údajů objednávky do XML
72 fwrite($fp, '
73     <dat:dataPackItem id="'. $row['order_id']. '" version="2.0">
74     <ord:order version="2.0">
75     <!--prijata objednavka s polozkama-->
76     <ord:orderHeader>
77     <ord:orderType>receivedOrder</ord:orderType>
78     <ord:numberOrder>'. $row["numberOrder_date"]. 'A'. $pocitadlo_text. '</ord:numberOrder>
79     <ord:date>'. $row["date_notime"]. '</ord:date>
80     <ord:dateFrom>'. $row["date_notime"]. '</ord:dateFrom>
81     <ord:dateTo>'. $row["date_notime"]. '</ord:dateTo>
82     <ord:text>'. substr ( $row["note"] , 0, 200 ). '</ord:text>
83     <ord:partnerIdentity>
84     <typ:address>
85     <typ:company>'. $address['firmname']. '</typ:company>
86     <typ:division>Obchodní oddělení</typ:division>
87     <typ:name>'. $address['name']. ' '. $address['surname']. '</typ:name>
88     <typ:city>'. $address['city']. '</typ:city>
89     <typ:street>'. $address['address']. '</typ:street>
90     <typ:zip>'. $address['zip']. '</typ:zip>
91     <typ:ico>'. $address['ico']. '</typ:ico>
92     <typ:dic>'. $address['dic']. '</typ:dic>
93     </typ:address>
94     </ord:partnerIdentity>
95     <ord:paymentType>
96     <typ:ids>'. $payment. '</typ:ids>
97     </ord:paymentType>
98     </ord:orderHeader>
99     <ord:orderDetail>');

```

Obr. 11: 3. část PHP skriptu pro import/export dat do Pohody (vlastní zpracování)

Nyní přejdeme k zápisu samotného XML kódu. Opět je zde použita funkce „fwrite“, se kterou zapíšeme řetězec znaků do souboru za pomoci proměnné \$fp. Všimněte si tagu <ord:numberOrder>, který má na starosti zápis čísla objednávky. Zápis čísla objednávky vypadá např. „20110205A001“, kde prvních osm znaků je datum objednávky, potom znak „A“ oddělující tři místné číslování od data objednávky. Dále si všimněte tagu <ord:text>, které má za úkol vypsát poznámku k objednavce. Musel zde být upraven string na 200 znaků z důvodu importu do Pohody, která hlásila chybu při delším textu 200 znaků.

Jelikož ovládám jazyk HTML a kaskádové styly, rozhodl jsem se, že vytvořím výstup na obrazovku s XML souborem připraveným pro export, aby práce splňovala estetická kritéria. Barvy stránky jsou použité z barev stránek firmy STORMWARE, která takto propaguje svůj produkt Pohodu. Výsledek (viz obrázek níže) je internetová stránka, která je HTML5 a CSS 2.1 validní dle konsorcia W3C.



[HTML5 validní](#) | [CSS 2.1 validní](#)

Obr. 12: Výstup obrazovky s požadovaným XML feedem (vlastní zpracování)

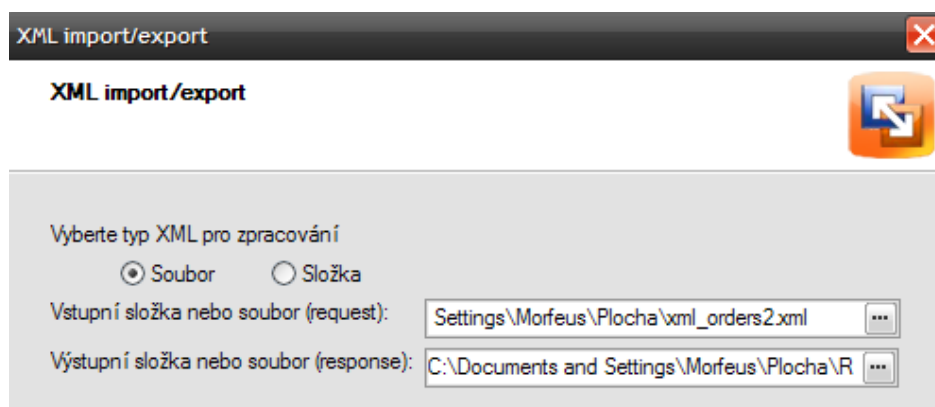
Pokud se takto (viz Obr. 12) zobrazí stránka, tak to znamená, že se PHP skript provedl úspěšně a že si bez problému můžeme zobrazit vygenerovaný XML soubor.

Pokud klikneme na odkaz, tak vygenerovaný soubor bude vypadat následovně:

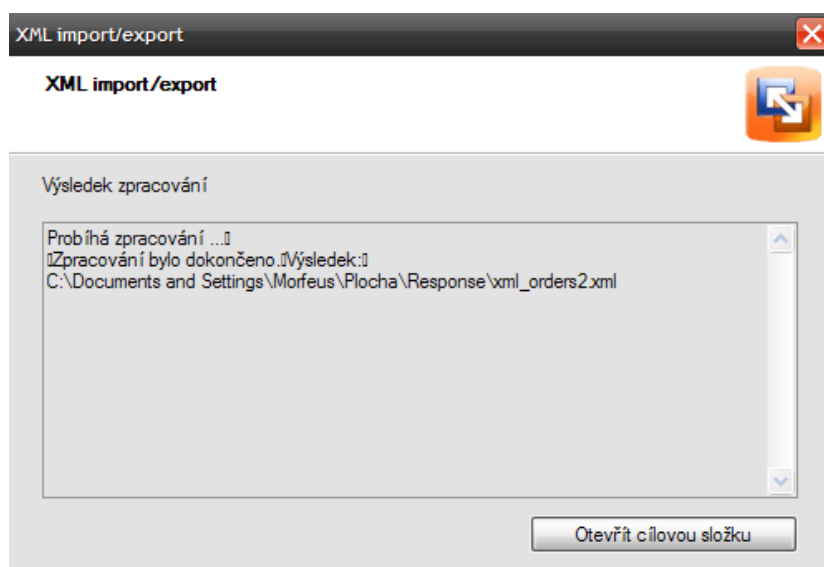
```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<dataPack xmlns:dat="http://www.stormware.cz/schema/version_2/data.xsd" xmlns:ord="http://www.stormware.cz/schema/version_2/order.xsd"
  xmlns:typ="http://www.stormware.cz/schema/version_2/type.xsd" id="ob001" ico="12345678" application="StwTest" version="2.0" note="Import Objednávky">
  <dat:dataPackItem id="2" version="2.0">
    <ord:order version="2.0">
      <!-- přijata objednávka s položkami -->
      <ord:orderHeader>
        <ord:orderType>receivedOrder</ord:orderType>
        <ord:numberOrder>20110204A001</ord:numberOrder>
        <ord:date>2011-02-04</ord:date>
        <ord:dateFrom>2011-02-04</ord:dateFrom>
        <ord:dateTo>2011-02-04</ord:dateTo>
        <ord:text/>
      </ord:orderHeader>
      <ord:partnerIdentity>
        <typ:address>
          <typ:company/>
          <typ:division>Obchodní oddělení</typ:division>
          <typ:name>Vit Sedláček</typ:name>
          <typ:city>Děhylov</typ:city>
          <typ:street>Forubská 43</typ:street>
          <typ:zip>74794</typ:zip>
          <typ:ico/>
          <typ:dic/>
        </typ:address>
        </ord:partnerIdentity>
        <ord:paymentType>
          <typ:ids>na dobírku</typ:ids>
        </ord:paymentType>
      </ord:orderHeader>
      <ord:orderDetail>
        <!-- textová položka -->
        <ord:orderItem>
          <ord:text>Kád BIDONE bílá 200 l</ord:text>
          <ord:quantity>1</ord:quantity>
          <ord:delivered>1</ord:delivered>
          <ord:rateVAT>high</ord:rateVAT>
          <ord:homeCurrency>
            <typ:unitPrice>$50.00</typ:unitPrice>
          </ord:homeCurrency>
        </ord:orderItem>
        <!-- textová položka -->
        <ord:orderItem>...</ord:orderItem>
      </ord:orderDetail>
      <ord:orderSummary>
        <ord:roundingDocument>mathZone</ord:roundingDocument>
      </ord:orderSummary>
    </ord:order>
  </dat:dataPackItem>
</dataPack>
```

Obr. 13: Část výsledného XML souboru pro Pohodu (vlastní zpracování)

Takto (viz Obr. 13) je soubor připraven pro import do programu Pohoda. Následně ho uložíme do počítače. Ještě předtím, než spustíme samotný import dat, projedeme tento kód validátorem, jestli je XML soubor v pořádku dle specifikace. Pokud ano, můžeme se pustit do závěrečné fáze, a tou je import dat, který nalezneme v programu Pohoda pod záložkou „Soubor > datová komunikace > XML import/export...“. Nyní se nám objeví tabulka, kde si vybereme soubor, který chceme importovat (viz Obr. 14).

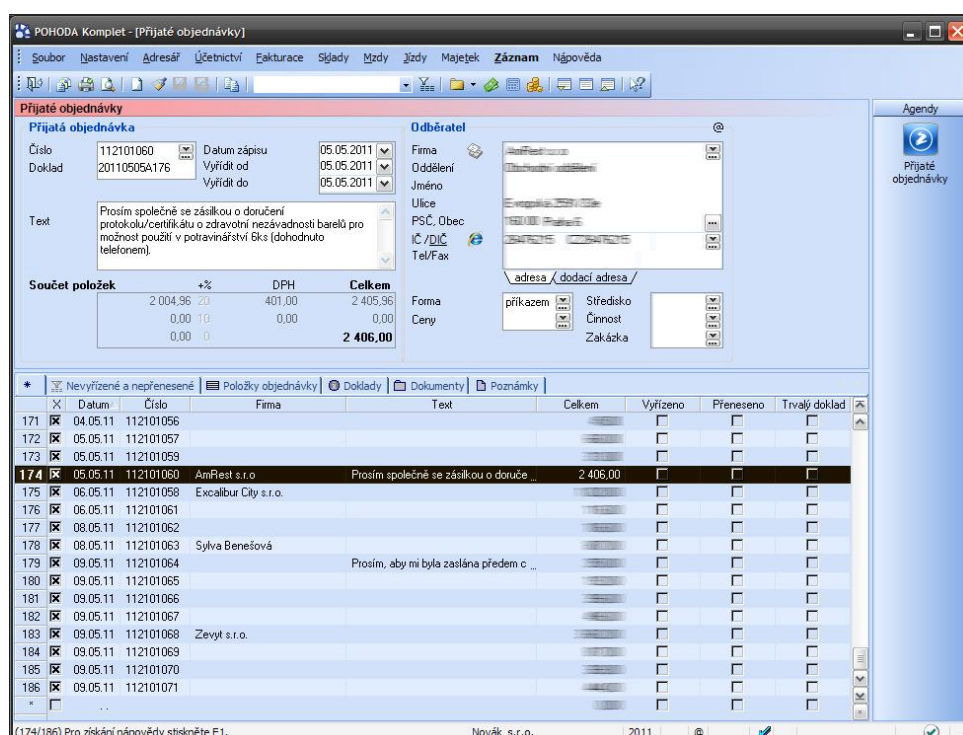


Obr. 14: Tabulka pro vložení XML souboru (vlastní zpracování)



Obr. 15: Výsledek zpracování XML importu/exportu (vlastní zpracování)

Pokud se zobrazí tabulka (viz Obr. 15), tak vše proběhlo v pořádku. Ještě si můžeme zkontrolovat tzv. LOG importu, kde se dozvíme, jestli se všechny objednávky importovaly v pořádku.



Obr. 16: Přehled importovaných objednávek v programu Pohoda (vlastní zpracování)

Konečná verze importu objednávek do programu Pohoda vypadá následovně (viz Obr. 16). Údaje jsou schválně rozmazané, abych nemohl poskytnout údaje firem, které objednávky provedly.

	Kód	Text	Množství	Přeneseno	M.j.	Jedn.cena	S	DPH	Sleva %	Částka	DPH
		Poznámka				Evidenční č.		Středisko		Činnost	Zakázka
1		Sud 120 l šroubovací ...	15,00	0,00		304,00	<input type="checkbox"/>	20 %	0,00%	4 560,00	912,00
2		Sud 60 l šroubovací s ...	10,00	0,00		239,00	<input type="checkbox"/>	20 %	0,00%	2 390,00	478,00
3		Kanystř 10 l s nálevko ...	5,00	0,00		57,00	<input type="checkbox"/>	20 %	0,00%	285,00	57,00
4		Kanystř 15 l s nálevko ...	5,00	0,00		79,00	<input type="checkbox"/>	20 %	0,00%	395,00	79,00
5		Kanystř 20 l s nálevko ...	5,00	0,00		89,00	<input type="checkbox"/>	20 %	0,00%	445,00	89,00
6		Spona na sud 120 l	10,00	0,00		67,00	<input type="checkbox"/>	20 %	0,00%	670,00	134,00
7		Spona na sud 220 l	10,00	0,00		80,00	<input type="checkbox"/>	20 %	0,00%	800,00	160,00
8		CLIVIA 20 květináč ža ...	5,00	0,00		10,00	<input type="checkbox"/>	20 %	0,00%	50,00	10,00
9		CLIVIA 25 květináč ža ...	5,00	0,00		17,00	<input type="checkbox"/>	20 %	0,00%	85,00	17,00
10		CLIVIA 30 květináč ža ...	5,00	0,00		27,00	<input type="checkbox"/>	20 %	0,00%	135,00	27,00
11		CLIVIA 35 květináč ža ...	7,00	0,00		36,00	<input type="checkbox"/>	20 %	0,00%	252,00	50,40
12		CLIVIA 50 květináč ža ...	5,00	0,00		62,00	<input type="checkbox"/>	20 %	0,00%	310,00	62,00

Obr. 17: Importované jednotlivé položky v programu Pohoda (vlastní zpracování)

Na obrázku (viz Obr. 17) jsou importované jednotlivé položky objednaného zboží. Jak si můžeme všimnout, je zde vidět název produktu, objednané množství, jednotná cena, DPH a pak celková částka bez DPH a částka DPH.

V analytické části byl jednoduše vysvětlený proces automatizace, který je zakreslen na Obr. 3. Tento proces automatizace se dá provést i pomocí tzv. CRONu, neboli plánovače úloh.

CRON je softwarový démon⁶, který slouží pro automatické spouštění procesů, nebo libovolných skriptů. Tyto libovolné skripty se mohou nacházet v adresáři vašeho počítače nebo na příslušném serveru. CRON se dá spouštět v operačních systémech, nebo případně na serveru, na kterém běží **webhosting**. V době psaní této práce firma TREF Josef Trojan měla své stránky pod správou webhostingové společnosti Blueboard.cz s.r.o. Tato společnost nabízí 2 varianty webhostingu, kde v té dražší (modré) variantě nabízí právě zmíněnou službu CRON. Abychom takovou službu mohli použít, je třeba znát přístupové údaje do klientské sekce webhostingu TREF.cz.

Při vytváření plánované úlohy je třeba dopředu vědět, který skript chceme, aby se pravidelně spouštěl. Dále je třeba nastavit časové rozmezí, případně, v kterých dnech chceme, aby se tento skript spouštěl. Nakonec je nutné nastavit časový interval. V našem případě jsem nastavil interval spouštění na 15 minut, což znamená, že každých 15 minut se spustí daný skript, který vygeneruje případné nové objednávky. Nesmí se zapomenout na nastavení práv souboru obsahující PHP skript. Tomuto souboru je třeba přiřadit minimálně práva čtení a spouštění vlastníkem, v osmičkové soustavě je to hodnota 500.

Automatický plánovač úloh (CRON) má velký přínos v tom, že nemusíme sami provádět skript, ale vše se děje automaticky. V našem případě se dá použít jak pro automatické generování XML dat pro Pohodu, tak i pro aut. generování XML dat pro server Zboží.cz, které funguje na stejném principu. Automatickým generováním dat si zajistíme aktuálnost informací o zboží na vyhledávacích serverech. V následující části této práce si uvedeme další přínosy návrhu řešení.

⁶ (CRON, In wikipedia, 2011)

4.3 Přínos návrhu řešení

V současné době má jazyk XML v internetovém světě velký potenciál a slibnou budoucnost, už jenom proto, že ho využívá stále více internetových aplikací a v konečném důsledku i samotných uživatelů (potenciálních zákazníků). Tato technologie disponuje vyšší bezpečností při komunikaci, kterou jsme si popsali v analýze problému. Je proto velkým přínosem a výčet jejího nasazení je obrovský.

Nasazením této technologie do firemního prostředí jsem firmě přinesl užitek jak časový, tak i ekonomický. Jako časový užitek mám na mysli proces automatizace, kde firmě ušetřím spoustu času a práce s ručním zapisováním, které by mohlo zabrat až několik týdnů. Napojení el. obchodu na IS není sice žádnou novinkou, přesto tato služba není zdarma. Pokud by si naše firma vyžádala profesionální řešení importu/exportu dat, propojení různých agend s IS, potom by to firmu v konečném důsledku mohlo stát až několik tisíc korun. Z ekonomického hlediska jsem proto firmě ušetřil spoustu finančních prostředků, pomocí nichž by si musela zajistit externistu, který by se jí staral o přepis operací mezi těmito IS.

Dalším důležitým přínosem bylo přidání zboží do katalogu vyhledávacího portálu Zboží.cz, kde by firma zjistila, zdali je cena jejího zboží konkurenceschopná. Na základě těchto údajů se firma dozvěděla skutečnost, že má jedny z nejnižších cen svého zboží na serveru Zboží.cz.

Závěr

Hlavním cílem této práce byl návrh úspěšné implementace XML pro vyhledávač zboží portálu Zboží.cz, a taky úspěšné provedení importu/exportu dat mezi el. obchodem TREF Josef Trojan a IS Pohoda od firmy STORMWARE. Nyní již s jistotou mohu tvrdit, že se mi hlavní cíl práce, který jsem si stanovil, úspěšně podařil dovést do konce.

Kromě již zmíněných přínosů má tato práce praktický užitek nejen pro firmu, ale i pro mě samotného. Díky ní jsem poznal, co vlastně jazyk XML znamená, jak tento jazyk použít ve prospěch firmy, a jak ho úspěšně implementovat. Dále jsem se naučil základům jazyka PHP a databáze MySQL. Největším přínosem pro mě bylo rozhodně vyřešení mnoha problémů, se kterými jsem se setkal v průběhu práce. Důležitým aspektem, který vedl ke zdárnému dokončení této práce, bylo dodržení specifikace a standardů webových služeb.

Díky této bakalářské práci jsem si prohloubil své znalosti dané problematiky, která mě nadchla natolik, že se jí do budoucna chci věnovat důkladněji.

Seznam použité literatury

Knižní zdroje

- [1] BASL, Josef. *Podnikové informační systémy : podnik v informační společnosti*. 2., výrazně přeprac. a rozš. vyd. Praha : Grada, 2008. 283 s. ISBN 978-802-4722-795.
- [2] KOFLER, M. *PHP 5 a MySQL 5 : průvodce webového programátora*. 1. vydání. Brno : Computer Press, 2007. 607 s. ISBN 978-80-251-1813-9.
- [3] MLÝNKOVÁ, Irena, et al. *XML technologie: principy a aplikace v praxi*. 1. vydání. Praha: Grada Publishing a.s. , 2008. 272 s. ISBN 9788024727257.
- [4] SODOMKA, Petr; KLČOVÁ, Hana. *Informační systémy v podnikové praxi*. 2. aktualiz. a rozš. vyd. Brno : Computer Press, 2010. 501 s. ISBN 978-802-5128-787.

Elektronické zdroje

- [5] BENDA, Pavel. *Webové služby : opatrný optimismus*. Systems Integration [online]. 2003, [cit. 2011-05-30]. Dostupné z WWW: <<http://si.vse.cz/archive/proceedings/2003/webove-sluzby-opatrný-optimismus.pdf>>.
- [6] Cron. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 16.11.2008, naposledy upraveno 24.4.2011 [cit. 2011-05-30]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Cron>>.
- [7] KOSEK, Jiří. *XML* [online]. 1999 [cit. 2011-05-18]. XML. Dostupné z WWW: <<http://www.kosek.cz/clanky/xml/>>.
- [8] SOLDÁT, Marek. *Slabikář XML - úvod do problematiky | Interval.cz* [online]. 25.3.2002 [cit. 2011-05-18]. Slabikář XML - úvod do problematiky. Dostupné z WWW: <<http://interval.cz/clanky/slabikar-xml-uvod-do-problematiky/>>.
- [9] STORMWARE. *Stormware software development* [online]. 2007 [cit. 2011-05-18]. Ekonomický a informační systém POHODA 2011. Dostupné z WWW: <<http://www.stormware.cz/pohoda/>>.

Seznam obrázků

OBR. 1: KOMUNIKACE MEZI KLIENTSKOU APLIKACÍ A WEBOVOU SLUŽBOU	12
OBR. 2: ZÁKLADNÍ STRUKTURA XML PRO VYHLEDÁVAČ ZBOŽÍ.CZ (NÁPOVĚDA SEZNAM.CZ)	28
OBR. 3: OBECNÝ MECHANISMUS AUTOMATIZOVANÉHO XML IMPORTU/EXPORTU (STORMWARE, 2011).....	31
OBR. 4: SCHÉMA DATABÁZE FIRMY TREF JOSEF TROJAN (VLASTNÍ ZPRACOVÁNÍ)	34
OBR. 5: 1. ČÁST PHP SKRIPTU PRO VYHLEDÁVAČ ZBOŽÍ.CZ (VLASTNÍ ZPRACOVÁNÍ).....	35
OBR. 6: 2. ČÁST PHP SKRIPTU PRO VYHLEDÁVAČ ZBOŽÍ.CZ (VLASTNÍ ZPRACOVÁNÍ).....	36
OBR. 7: ČÁST VÝSLEDNÉHO XML SOUBORU (VLASTNÍ ZPRACOVÁNÍ)	37
OBR. 8: VÝSLEDEK VYHLEDÁVÁNÍ ZBOŽÍ NA SERVERU ZBOŽÍ.CZ (VLASTNÍ ZPRACOVÁNÍ)	38
OBR. 9: 1. ČÁST PHP SKRIPTU PRO IMPORT/EXPORT DAT DO POHODY (VLASTNÍ ZPRACOVÁNÍ)	39
OBR. 10: 2. ČÁST PHP SKRIPTU PRO IMPORT/EXPORT DAT DO POHODY (VLASTNÍ ZPRACOVÁNÍ)	40
OBR. 11: 3. ČÁST PHP SKRIPTU PRO IMPORT/EXPORT DAT DO POHODY (VLASTNÍ ZPRACOVÁNÍ)	41
OBR. 12: VÝSTUP OBRAZOVKY S POŽADOVANÝM XML FEEDEM (VLASTNÍ ZPRACOVÁNÍ)	42
OBR. 13: ČÁST VÝSLEDNÉHO XML SOUBORU PRO POHODU (VLASTNÍ ZPRACOVÁNÍ)	43
OBR. 14: TABULKA PRO VLOŽENÍ XML SOUBORU (VLASTNÍ ZPRACOVÁNÍ).....	43
OBR. 15: VÝSLEDEK ZPRACOVÁNÍ XML IMPORTU/EXPORTU (VLASTNÍ ZPRACOVÁNÍ)	44
OBR. 16: PŘEHLED IMPORTOVANÝCH OBJEDNÁVEK V PROGRAMU POHODA (VLASTNÍ ZPRACOVÁNÍ).....	44
OBR. 17: IMPORTOVANÉ JEDNOTLIVÉ POLOŽKY V PROGRAMU POHODA (VLASTNÍ ZPRACOVÁNÍ)	45

Seznam tabulek

TAB. 1: ZÁKLADNÍ VESTAVĚNÉ DATOVÉ TYPY (MLÝNKOVÁ, 2008, s. 53).....	23
TAB. 2: VÝZNAM ZNAČEK PRO SERVER ZBOŽÍ.CZ (NÁPOVĚDA SEZNAM.CZ)	29

Seznam použitých zkratek a pojmů

APS (Advanced Planning and Scheduling) – systémy pokročilého rozhodování a plánování výroby

Company Information system (CIS) – podnikové informační systémy

CRM (Customer Relationship Management) – řízení vztahů se zákazníky

DTD (Document Type Definition) – definice typu dokumentu

ERP (Enterprise Resource Planning) – plánování a řízení firemních zdrojů, typ podnikového IS schopného pokrýt a integrovat klíčové procesy ve firmě.

FTP přístup (File Transfer Protocol) – jedná se o přístup na server, pomocí kterého dochází k přenosu dat mezi PC a serverem v počítačové síti

Webhosting – je služba, která zajišťuje vlastní místo na internetu

Mainframe – výkonný PC (mnohdy až sálové PC), které se zabývají zpracováním velkého objemu dat

MIS (Management Information System) – samostatný manažerský informační systém nebo modulární součást ERP systému, podporující rozhodovací procesy v podniku

PHP (rekurzivní značka Hypertext Preprocessor) – skriptovací programovací jazyk

Proklik – znamená kliknutí na odkaz, pomocí kterého se dostaneme na cílovou stránku (oproti běžnému odkazu se jedná o reklamní odkaz, většinou placený)

RIA (Rich Internet Application)

SCM (Supply Chain Management) – řízení dodavatelských řetězců

SOA (Service Oriented Architecture) – servisně orientovaná architektura

Tag – vymezení smyslu dokumentu a jeho vzhledu

URL (Uniform Resource Locator) – v překladu jednotný lokátor zdrojů

Validátor – kontrolor chyb, který se řídí různými specifikacemi dokumentu

Webové služby (Web services)

XML (eXtensible Markup Language) – rozšiřitelný značkovací jazyk

XML feed – soubor, ze kterého získáváme informace o nabízených položkách

Seznam příloh

Příloha č.1 - PHP skript pro vyhledávač zboží Zboží.cz

```
<?php
// natažení knihoven eshopu
require_once("../functions/global_fns.php");
require_once("../functions/eshop_fns.php");

// připojení k databázi
$conn = db_connect();

// výběr všech produktů z databáze
$query = "SELECT * FROM products";

// provedení dotazu
$result = $conn->query($query);

// vytvoření otevření souboru, který je připraven pro zapis XML
$fp = fopen('xml_feed_zbozi.xml', 'w');

fwrite($fp, "<?xml version=\"1.0\" encoding=\"windows-1250\"?>\r\n");
fwrite($fp, "<SHOP>\r\n\r\n");

// vybrání řádku z tabulky
while($row = $result->fetch_assoc()) {
    extract($row);
    if($description == NULL || !$depict)
        continue;

    $subcategory = get_subcategory_info($subcategory_id);
    $category = get_category_info($subcategory['category_id']);
    if(!$category || !$subcategory)
        continue;

    $count <= 0 ? $availability = "-1" : $availability = "0";

    fwrite($fp, "<SHOPITEM>\r\n");
    fwrite($fp, "<PRODUCT>".$product_name."</PRODUCT>\r\n");
    fwrite($fp, "<DESCRIPTION>".$product_name."</DESCRIPTION>\r\n");
    fwrite($fp, "<URL>http://www.tref.cz/product_detail.php?prodid=".$product_id."</URL>\r\n");
    fwrite($fp, "<ITEM_TYPE>new</ITEM_TYPE>\r\n");
    fwrite($fp, "<DELIVERY_DATE>".$availability."</DELIVERY_DATE>\r\n");
    fwrite($fp,
        "<IMGURL>http://tref.cz/pictures/".$pictures_dir."/thumbs/thumb_".str_
        replace(" ", "%20", $picture)."</IMGURL>\r\n");
    fwrite($fp, "<PRICE_VAT>".$price."</PRICE_VAT>\r\n");
    fwrite($fp, "<CATEGORYTEXT>".$category["category_name"]." |
        ".$subcategory["subcategory_name"]."</CATEGORYTEXT>\r\n");
    fwrite($fp, "</SHOPITEM>\r\n\r\n");
}
fwrite($fp, "</SHOP>\r\n");
?>
```

Příloha č.2 - PHP skript pro import/export dat z el. obchodu do Pohody

```
<?php

// natažení knihoven eshopu
require_once("../functions/global_fns.php");
require_once("../functions/eshop_fns.php");

// připojení k databázi
$conn = db_connect();

// výběr všech objednávek z databáze
$query = "SELECT *,
DATE_FORMAT(date, '%Y%m%d') AS numberOrder_date,
DATE_FORMAT(date, '%Y-%m-%d') AS date_notime
FROM orders";
$result = $conn->query($query);

// vytvoření deskriptoru souboru pro zápis XML
$fp = fopen('xml_orders2.xml', 'w');

// zápis hlaviček xml dokumentu
fwrite($fp, '<?xml version="1.0" encoding="Windows-1250"?>
<dat:dataPack id="ob001" ico="12345678" application="StwTest" version
= "2.0" note="Import Objednávky"
xmlns:dat="http://www.stormware.cz/schema/version_2/data.xsd"
xmlns:ord="http://www.stormware.cz/schema/version_2/order.xsd"
xmlns:typ="http://www.stormware.cz/schema/version_2/type.xsd">');

// inicializace pomocných proměnných
$pocitadlo = 1;

// zpracování objednávek
while($row = $result->fetch_assoc()) {

// převod počítadla objednávek na 3-místné číslo
if($pocitadlo < 10)
    $pocitadlo_text = "00".$pocitadlo;
elseif($pocitadlo < 100)
    $pocitadlo_text = "0".$pocitadlo;
else
    $pocitadlo_text = $pocitadlo;

// převod způsobu dopravy na česke ekvivalenty
switch($row['payout']) {
case "cod":
    $payment = "na dobírku";
break;
case "cash":
    $payment = "hotově";
break;
case "transfer":
    $payment = "převodem";
}
// převod proznaku stavu objednávky
```

```

switch($row['status']) {
case "done":
$delivered = "1";
break;
case "pending":
$delivered = "0";
}
// výběr adresy fakturačních dat objednávky z databáze
$sql = "SELECT *
FROM address_data
WHERE address_id = ".$row['invoice_id'];

$address_result = $conn->query($sql);
$address = $address_result->fetch_assoc();

// zápis údajů objednávky do XML
fwrite($fp, '
<dat:dataPackItem id="'.$row['order_id'].'" version="2.0">
<ord:order version="2.0">
<!--prijata objednavka s polozkama-->
<ord:orderHeader>
<ord:orderType>receivedOrder</ord:orderType>
<ord:numberOrder>'.$row["numberOrder_date"].'A'.$pocitadlo_text.'</ord:
numberOrder>
<ord:date>'.$row["date_notime"].'</ord:date>
<ord:dateFrom>'.$row["date_notime"].'</ord:dateFrom>
<ord:dateTo>'.$row["date_notime"].'</ord:dateTo>
<ord:text>'.substr ( $row["note"] , 0, 200 ).'</ord:text>
<ord:partnerIdentity>
<typ:address>
<typ:company>'.$address['firmname'].'</typ:company>
<typ:division>Obchodní oddělení</typ:division>
<typ:name>'.$address['name'].' '.$address['surname'].'</typ:name>
<typ:city>'.$address['city'].'</typ:city>
<typ:street>'.$address['address'].'</typ:street>
<typ:zip>'.$address['zip'].'</typ:zip>
<typ:ico>'.$address['ico'].'</typ:ico>
<typ:dic>'.$address['dic'].'</typ:dic>
</typ:address>
</ord:partnerIdentity>
<ord:paymentType>
<typ:ids>'.$payment.'</typ:ids>
</ord:paymentType>
</ord:orderHeader>
<ord:orderDetail>');

// výběr objednaných produktů objednávky z databáze
$sql = "SELECT order_products.*, products.product_name
FROM order_products, products
WHERE order_products.product_id = products.product_id
AND order_products.order_id = ".$row['order_id']."";
$products_result = $conn->query($sql);

// zpracování produktu
while($product = $products_result->fetch_assoc()) {

// zápis objednaných produktů objednávky do XML
fwrite($fp, '
<!--textova polozka-->

```

```

<ord:orderItem>
<ord:text>'.$product['product_name'].</ord:text>
<ord:quantity>'.$product['quantity'].</ord:quantity>
<ord:delivered>'.$delivered.</ord:delivered>

<ord:rateVAT>high</ord:rateVAT>
<ord:homeCurrency>
<typ:unitPrice>'.$product['product_price'].</typ:unitPrice>
</ord:homeCurrency>
</ord:orderItem>');

// výpočet celkové ceny objednávek
$celkova_cena_objednavek += $product['product_price'] *
$product['quantity'];

// další údaje
$id_objednavky = $product['order_id'];
$id_produkту = $product['product_id'];
$jmeno_produkту = $product['product_name'];
$scena_produkту = $product['product_price'];
$pocet_produkту = $product['quantity'];
$barva_produkту = $product['color'];
}
$pocitadlo++;
// zápis XML zápatí
fwrite($fp, '
</ord:orderDetail>
<ord:orderSummary>
<ord:roundingDocument>math2one</ord:roundingDocument>
</ord:orderSummary>
</ord:order>
</dat:dataPackItem>');
}
fwrite($fp, '</dat:dataPack>');

// výpis přehledu zpracování na výstup obrazovky
echo ('<!DOCTYPE HTML>
<html>
<head>
<title>Bakalarska prace - EXPORT OBJEDNAVEK do Pohody</title>
<meta http-equiv="content-type" content="text/html; charset=windows-
1250" />
<meta name="description" content="Využití XML v informačním systému
firmy TREF.CZ" />
<meta name="robots" content="nofollow" />
<meta name="keywords" content="Bakalářská práce, Export dat do Pohody,
XML" />
<meta name="author" content="Ondřej Bachurek |
xbachu00@stud.fbm.vutbr.cz" />
<meta name="copyright" content="Ondřej Bachurek" />
<link rel="stylesheet" type="text/css"
href="http://tref.cz/admin/css1/layout.css" media="screen, projection"
/>
<link rel="stylesheet" type="text/css"
href="http://tref.cz/admin/css1/base.css" media="screen, projection"
/>
</head>
<body>
<h1>EXPORT OBJEDNÁVEK

```



```

do Pohody</h1>
<div id="hlavni">
<div id="topmenu">
<h3>Export dat proběhl úspěšně, soubor naleznete <a title="XML soubor
objednávek" href="xml_orders2.xml">zde</a></h3>
</div>
<div id="main">
<p>
<a title="HTML validator"
href="http://validator.w3.org/check?uri=referer">HTML5 validní</a> |
<a title="CSS validator" href="http://jigsaw.w3.org/css-
validator/check/referer">CSS 2.1 validní</a>
</p>
</div></div>
</body></html>' );?>

```